Final Report on

NASA CR-
144410

THE DESIGN OF A DIGITAL VOICE DATA COMPRESSION TECHNIQUE FOR ORBITER VOICE CHANNELS

CONTRACT NO. NAS 9-14416

TRW NO. 27165
AUGUST 1975

Prepared for

LYNDON B. JOHNSON SPACE CENTER
Houston, Texas

TRW
SYSTEMS GROUP

ONE SPACE PARK REDONDO BEACH, CALIFORNIA 90278

TABLE OF CONTENTS

1 INTRODUCTION

1.1 BACKGROUND

This study of voice bandwidth compression techniques was motivated by anticipated link margin difficulties in the Shuttle S-band communication system. It was felt that by reducing the data rate on each voice channel from the baseline 24 (or 32) Kbps to 8 Kbps, additional margin could be obtained. Thus, this study was undertaken to determine the feasibility of such an alternate voice transmission system. Several factors of prime importance that were addressed are:

1) Achieving high quality voice at 8 Kbps,

2) performance in the presence of the anticipated shuttle cabin environmental noise,

3) performance in the presence of the anticipated channel error statistics,

4) minimal increase in size, weight, and power over the current baseline voice processor.

1.2 TASKS

The following is a summary of the tasks performed under this contract:

Task 1: System Requirement Analysis

Descriptions of the operational environment including cabin background noise and channel error rates were determined through consultation with NASA-JSC personnel. Material for system testing was supplied by NASA/JSC. It was decided, based on previous work at TRW, to examine various forms of adaptive predictive coding (APC) for the Orbiter application.

Task 2: Compression Technique Design and Evaluation

Preliminary analysis of several existing APC algorithms led to the selection of a computationally simple algorithm with a noise squelching function. This basic algorithm which was developed under a TRW IR&D program was then subjected to a preliminary parameter optimization specific to Shuttle requirements. The system was simulated in integer arithmetic on an Interdata 85 computer, and voice tapes provided by NASA/JSC were processed through the simulation. The properties of the proposed channel were also simulated and several tapes were processed through the APC system and simulated channel.

TASK 3: Parameter Optimization

Tradeoffs were performed varying the parameters of the chosen system, such as frame rate, sampling rate, number of coefficients, etc., and it was determined that the original system was near optimum in light of the implementation constraints. A revised baseline was established and used to process one tape. In consultation with NASA/JSC, it was decided to examine a special-purpose hardware implementation based on a micro-controller and arithmetic unit. A preliminary design and a sizing were then performed for this configuration.

1.3 SUMMARY

An 8 kbps Adaptive Predictive Coding system for potential orbiter use was designed and then simulated in integer arithmetic. The system performs well and shows good resistance to both channel errors and background noise similar to those anticipated in the orbiter application. Channel error rates ranging from 10^{-4} to 10^{-2} were simulated with the result that rates of 10^{-3} or less were judged to have negligible impact on the received voice quality, and a rate of 10^{-2}, while noticeable, produced no unexpected distortions. Due to the inclusion of an adaptive squelching unit, the level of noise in the received voice was often below the input noise level at the transmitter.

A preliminary estimate of the flight hardware configuration indicates that one full duplex system could be implemented in fewer than 200 C-MOS IC's (including both LSI and MSI chips) using less than 20 watts. The amount of hardware required is insensitive to small variations (ten percent or less) in both the data rate and frame size. Thus, the system is easily adaptable to minor changes in the Network Signal Processor operation.

2 VOICE ALGORITHM

2.1 THEORY

The chosen system is a form of adaptive delta modulation (Figure 2.1) in which both the quantizer step size and the predictor coefficients are optimized over a time interval (called a frame) for the current speech statistics. Experimental observations of speech statistics indicate that the speech signal can be considered stationary over a time interval of 10 to 30 milliseconds. Thus, a typical frame time is chosen in this range.

2.1.1 Transmitter

The predictor coefficients are chosen to minimize the power in the error, e_n, given by

$$e_n = S_n + \sum_{j=1}^{M} a_j \, S_{n-j} \tag{2.1}$$

Thus, the power for an N sample frame is

$$E = \sum_{n=1}^{N} e_n^2 = \sum_{n=1}^{N} (S_n + \sum_{j=1}^{M} a_j \, S_{n-j})^2 \tag{2.2}$$

E is minimized by setting $\partial E/\partial a_i = 0 \quad i = 1, \ldots , M$
which results in a system of equations

$$\sum_{n=1}^{N} S_n \, S_{n-i} + \sum_{j=1}^{M} a_j \sum_{n=1}^{N} S_{n-j} \, S_{n-i} = 0 \qquad i = 1, \ldots, M \tag{2.3}$$

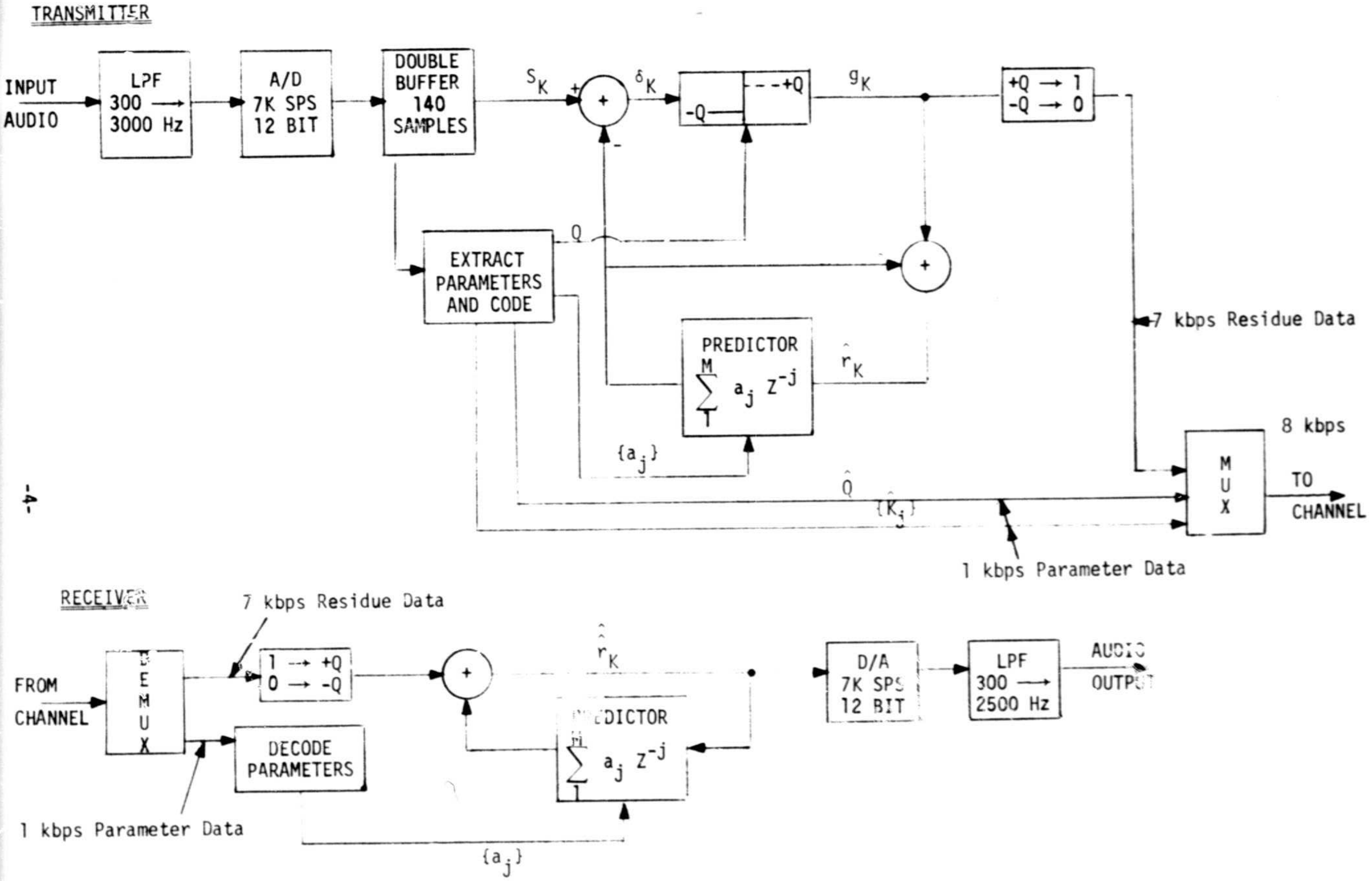

Figure 2.1 System Summary

Several schools of thought exist on the solution of this system. We have chosen the autocorrelation formulation, which yields the simplest solution. For this approach it is assumed that $S_n = 0$ for $n < 1$ and $n > N$. Thus,

$$\sum_{n=1}^{N} S_{n-j}\, S_{n-i} = \sum_{n=1}^{N-|j-i|} S_n\, S_{n+|j-i|} = R_{|j-i|},$$ i.e., each frame is considered independently from its neighbors. We may re-phrase the problem as follows:

(1) Compute $$R_i = \sum_{n=1}^{N-i} S_n\, S_{n+i}$$

(2) Solve $$\sum_{j=1}^{M} a_j\, R_{|j-i|} = -R_i \quad i = 1,\ldots, M$$

Clearly, the correlation coefficients $R_{|j-i|}$ form a matrix $[r_{ij}]$ which is symmetric, has all positive entries, and in which all the elements along the diagonal or any off-diagonal are equal. Further, the values $R_{|j-i|}$ are all chosen from the set $\{R_0,\ldots,R_{M-1}\}$. Systems of equations involving such a matrix (which is called Toeplitz) are easily solved using a technique known as Levinson's recursion[2]. A flowchart for Levinson's recursion is given in Figure 2.2.

Unfortunately, the solution of this system of equations may lead to a formulation of a filter which is unstable, that is, some of the zeros of

$$1 + \sum_{j=1}^{M} a_j\, Z^{-j}$$

or equivalently, some of the poles of

$$\frac{1}{1 + \sum_{j=1}^{M} a_j Z^{-j}}$$

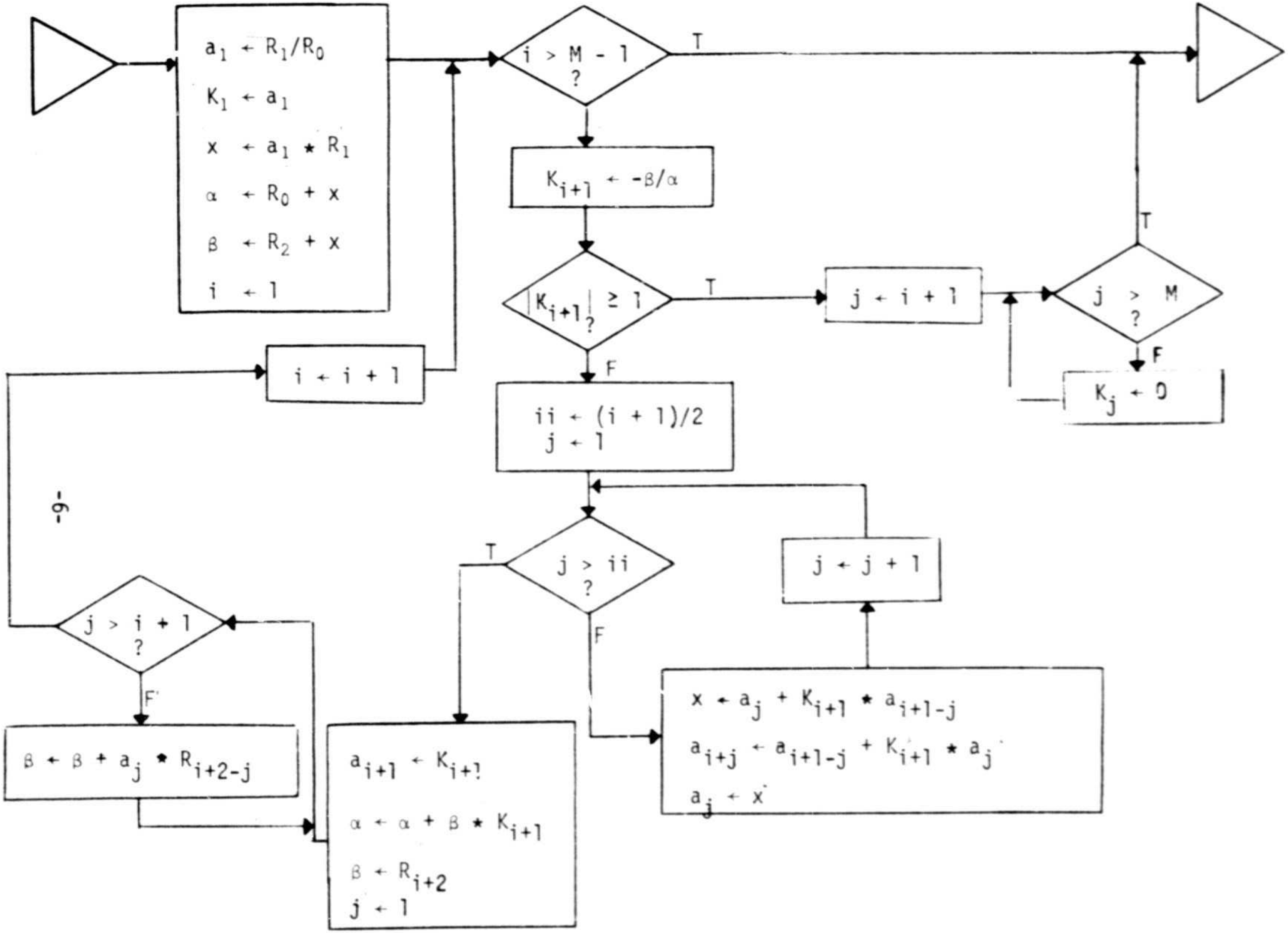

Figure 2.2 Levinson Recursion

may lie outside the unit circle. This is most likely to occur due to round off and truncation errors when the input speech has one or more components with narrow bandwidths. Thus, it is usual to window the frame of speech samples with a function which tapers to zero near the end points of the frame. A nearly equivalent approach which is computationally simpler, is to window the autocorrelation function in such a way as to provide a slight increase in bandwidth in the resulting filter. To a good approximation: If a signal $S(t)$ has bandwidth B and autocorrelation function $R(\tau)$, then the autocorrelation function $e^{-\alpha|\tau|}R(\tau)$ corresponds to a signal with bandwidth $B + 2\alpha$. The appropriate modification to digital autocorrelation coefficients is

$$\hat{R}_i = \exp\{-\pi FTi\}\ R_i$$

where T is the sampling interval in seconds, and F is the increase in bandwidth in Hz. Values of F in the range 10 to 50 Hz have been found experimentally to be sufficient to ensure stability. The Levinson recursion is then applied to the values $\{\hat{R}_i\}$.

A by-product of the Levinson recursion is a set of M coefficients $\{K_j\}_{j=1}^{M}$, known as the reflection coefficients. The properties of the reflection coefficients have been studied at length. They are bounded by ± 1 for stable filters, and thus provide a stability check. Because of this and the fact that a simple algorithm exists for transforming reflection coefficieints to predictor coefficients, it is clear that it is the reflection coefficients which should be quantized for transmission. However, it is important that the predictor coefficients used in both the transmitter and receiver be identical in the absence of channel errors. Thus, at the conclusion of the Levinson recursion, the reflection coefficients are quantized, and the quantized values are applied to a short form Levinson procedure (Figure 2.3) to produce a set of predictor coefficients for use in the transmitter's quantizer loop.

Another by-product of the Levinson recursion is the quantity, α, which measures the power of the prediction error. The rms prediction error per sample is $\langle e_n^2 \rangle^{1/2} = \sqrt{\alpha/N}$. This prediction error gives a good indication of the minimum quantization error $\langle (Q-|\delta_n|)^2 \rangle^{1/2}$.

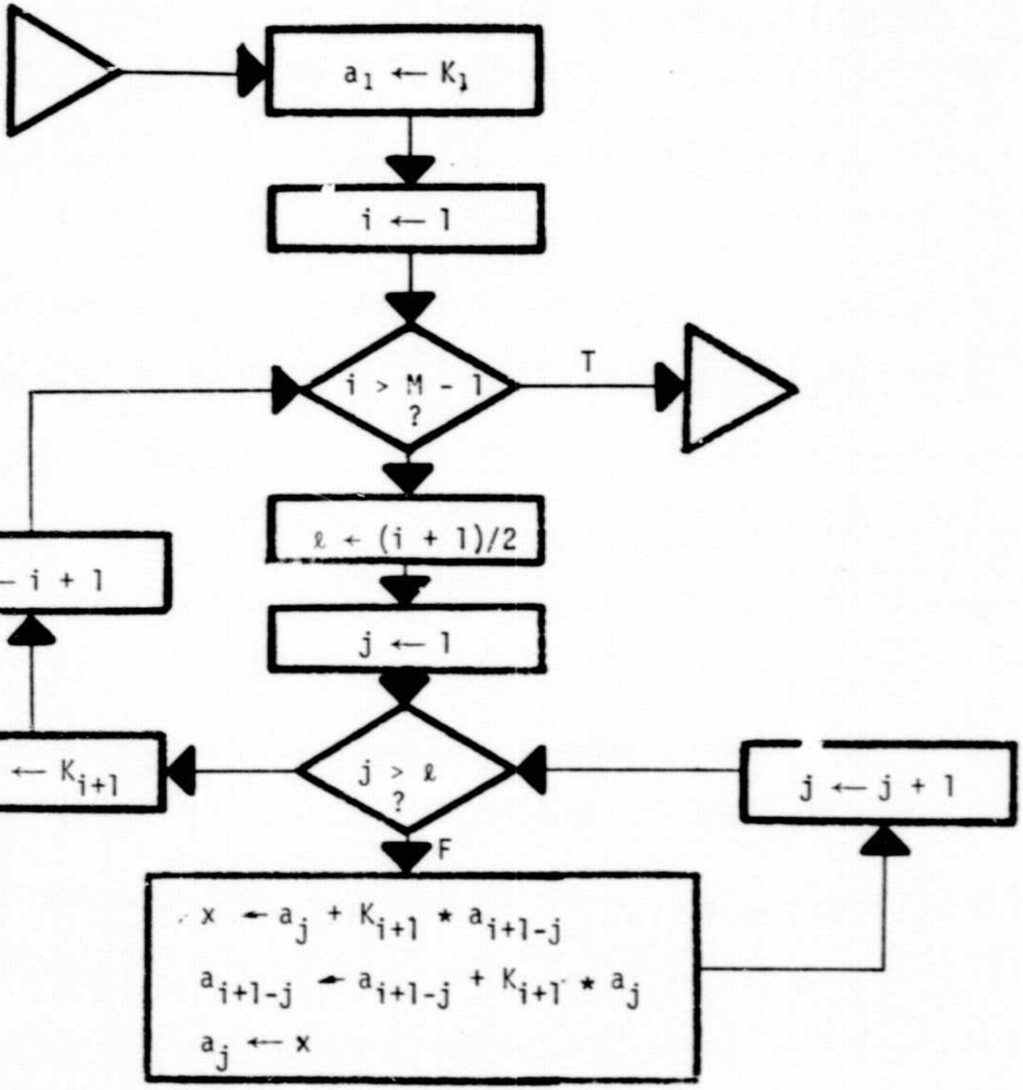
$a_1 \leftarrow K_1$
$i \leftarrow 1$
$i > M - 1$?
T
$\ell \leftarrow (i + 1)/2$
$j \leftarrow 1$
$j > \ell$?
F
$x \leftarrow a_j + K_{i+1} * a_{i+1-j}$
$a_{i+1-j} \leftarrow a_{i+1-j} + K_{i+1} * a_j$
$a_j \leftarrow x$
$j \leftarrow j + 1$
$a_{i+1} \leftarrow K_{i+1}$
$i \leftarrow i + 1$

Figure 2.3 Conversion of Reflection Coefficients to Predictor Coefficients

With reference to Figure 2.1, it can be shown that the optimum quantizer setting, Q, which minimizes quantization error, is given by

$$Q = \langle |\delta_n| \rangle$$

For most voice signals, it has been observed that for reasonable choices of quantizer level, Q,

$$\langle |\delta_n| \rangle \cong .7 \langle \delta_n^2 \rangle^{1/2}$$

As the system tracks the input signal, S_n, the signal $\hat{r}_n$ is forced to follow the input signal. Thus, the prediction error, δ_n, within the loop approximates the theoretical prediction error, e_n. So a good approximation to the quantizer level is

$$Q = .7 \langle \delta_n^2 \rangle^{1/2} \cong .7 \sqrt{\alpha/N}$$

It has been found experimentally that the ratio α/R_0 of the output power of the ideal predictor to the input power does not exceed 0.36 for voice signals. Thus, this ratio is also used in computing the quantizer level, i.e.,

$$Q = .7 \; F(\alpha/R_0) \; \sqrt{\alpha/N}$$

when α/R_0 approaches 1 indicating severe noise in the input, $F(\alpha/R_0)$ approaches zero, thus reducing the ability of the loop to track the signal.

In the loop, the output of the quantizer is fed back through the predictor thus providing a closed loop prediction system. In addition, the quantizer output (values of $\pm Q$ only) is converted to a bit stream called the residue at a rate of one bit per sample interval. The residue for each frame is multiplexed with the corresponding coded parameters to give the channel bit stream.

2.1.2 Receiver

In the receiver or demodulator, the residue bit stream is converted to a stream of ±Q values and used to drive a synthesis loop (see Figure 2.1). The predictor coefficients which are fixed over a frame, are obtained from the received reflection coefficients by the short Levinson recursion (Figure 2.3).

2.1.3 Parameter Encoding

The quantizer level has a wide dynamic range. Thus, it is coded logarithmically. Subjective listening indicated negligible difference between quantization to 4 bits and no quantization. This 4-bit coding scheme was used for all simulations and testing.

The reflection coefficients were coded linearly to 4 bits each. Again, subjective listening tests indicated little if any difference between this quantization scheme and no quantization. Other quantization techniques, such as log bilinear, have been shown to yield a slightly higher quality voice in other types of compression systems. It was felt, however, that the slight improvement that might be obtained here did not justify the additional computational complexity.

2.1.4 Baseline System

Preliminary subjective listening tests were used to select the following baseline system:

Sampling Rate: 7 KHz (12 bits/sample)

Frame Time: 20 ms

Parameters:

Q	Coded logarithmically to 4 bits
K_1	Coded linearly to 4 bits
K_2	Coded linearly to 4 bits
K_3	Coded linearly to 4 bits
K_4	Coded linearly to 4 bits
	Total: 20 bits/frame

Data Rate:	Parameters	1 Kbps
	Residue	7 Kbps
	Total	8 Kbps

This is the system that was used for simulation and testing. Detailed tradeoffs of these parameter values are discussed in Section 2.4.

In practice, several bits per frame could be devoted to frame synch. A typical allocation with provision for frame synch is:

Sampling Rate:	6750
Frame Time:	20 ms
Parameters:	20 bits/frame
Synch:	5 bits/frame

2.1.5 Channel Characteristics

In addition to resistance to background noise in the speaker's environment, the resistance of the system to channel errors is of importance. In order to determine the effects of the proposed channel, the statistics of burst errors resembling those of a rate 1/3 Viterbi decoder with constraint length 7 were developed for error rates of 10^{-2}, 10^{-3}, and 10^{-4}. Appendix A describes the channel simulation techniques in detail.

2.2 INTEGER SIMULATION OF NASA-APC ALGORITHMS ON INTERDATA 85

An integer simulation of the NASA-APC algorithms was developed on the Interdata 85, a 16-bit machine, in two versions:

IAPCDC:	Disk-to-disk I/O
IAPCTC:	Tape-to-tape I/O.

The programs are identical except for the I/O subroutines called.

The programs execute NASA-APC transmitter and receiver algorithms, and simulate burst errors (as specified by parameter input) on the communication channel.

The main programs are in FORTRAN and use functions and subroutines to simulate fractional integer arithmetic. The simulation equated + 1 to 32767 and expressed fractions as a proportion of the + 1 base.

The programs require the following parameters during initialization:

M	Number of Coefficients
SR	Sampling Rate
NSPF	Number of Samples per Frame
ERRATE	Channel Error Flag +, Simulate channel errors -, Do not simulate channel errors
FRAC	Quantization Level Gain Control
F	Quantization Level Noise Squelch Control
JAQ_i	Reflection Coefficient Quantization, i = 1, M
JREC	Number of Input Frames to Skip Before Processing
NUMREC	Number of Frames to Process -, Process to end-of-file
NPC	Maximum Channel Burst Length
EC	Channel: E_s/N_0
PC_i	Probability of Burst of Length i, i = 1, NPC
IT_i	Input Seeds to Random Number Generator i = 1, 19

At the end of initialization, the programs inform the operator of the bit rate. If channel errors are simulated, the program prints the total number of errors simulated every 100 frames. When the number of frames designated has been processed, the program prints the number of uses of each burst length and the calculated error rate. A flowchart of the program is given in Figure 2.4.

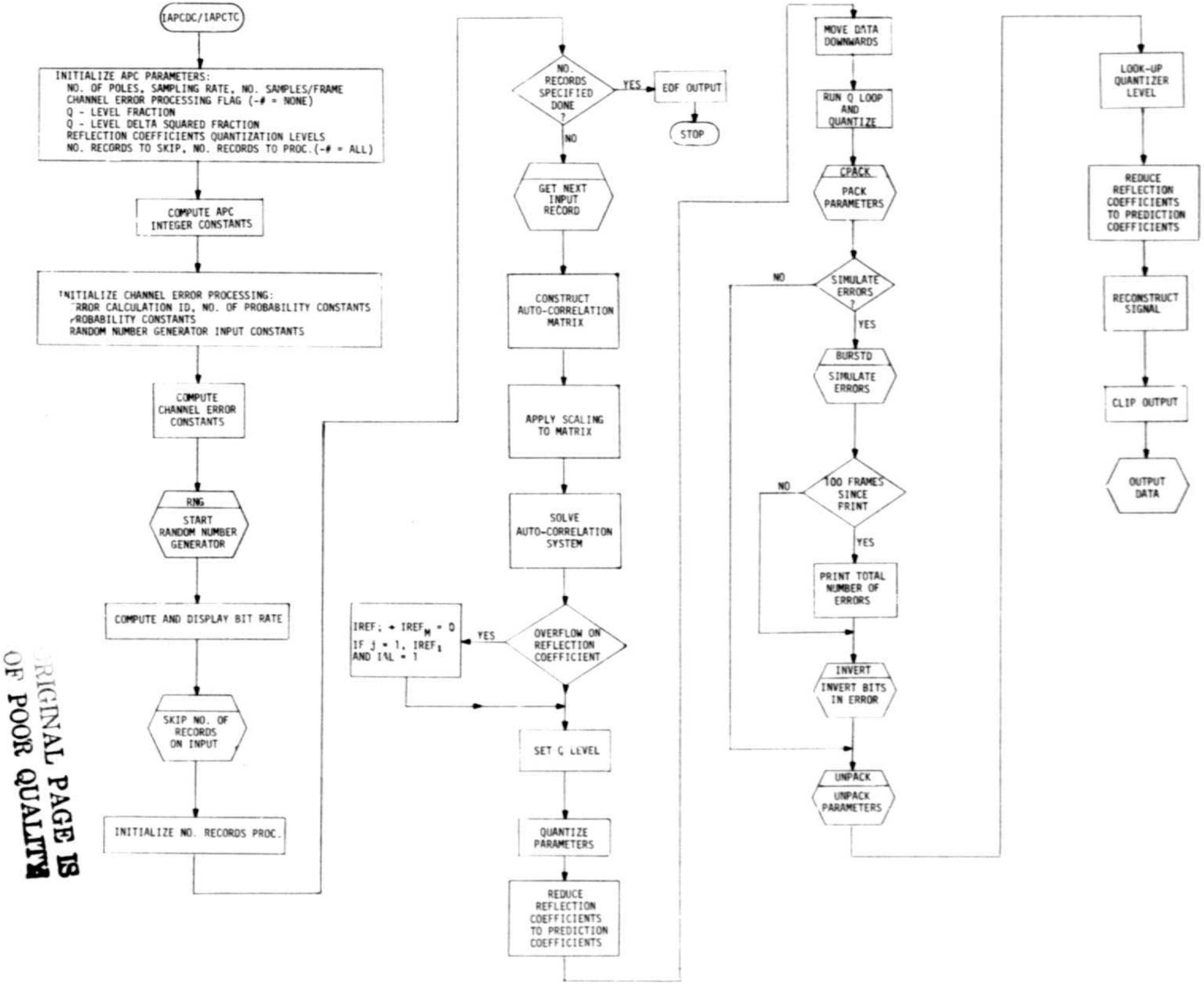
IAPCDC/IAPCTC
INITIALIZE APC PARAMETERS:
NO. OF POLES, SAMPLING RATE, NO. SAMPLES/FRAME
CHANNEL ERROR PROCESSING FLAG (-# = NONE)
Q - LEVEL FRACTION
Q - LEVEL DELTA SQUARED FRACTION
REFLECTION COEFFICIENTS QUANTIZATION LEVELS
NO. RECORDS TO SKIP, NO. RECORDS TO PROC. (-# = ALL)
COMPUTE APC INTEGER CONSTANTS
INITIALIZE CHANNEL ERROR PROCESSING:
ERROR CALCULATION ID, NO. OF PROBABILITY CONSTANTS
PROBABILITY CONSTANTS
RANDOM NUMBER GENERATOR INPUT CONSTANTS
COMPUTE CHANNEL ERROR CONSTANTS
RNG
START RANDOM NUMBER GENERATOR
COMPUTE AND DISPLAY BIT RATE
SKIP NO. OF RECORDS ON INPUT
INITIALIZE NO. RECORDS PROC.
NO. RECORDS SPECIFIED DONE ?
YES
EOF OUTPUT
STOP
NO
GET NEXT INPUT RECORD
CONSTRUCT AUTO-CORRELATION MATRIX
APPLY SCALING TO MATRIX
SOLVE AUTO-CORRELATION SYSTEM
OVERFLOW ON REFLECTION COEFFICIENT
YES
IREF$_i$ + IREF$_M$ = 0
IF j = 1, IREF$_1$
AND I%L = 1
SET Q LEVEL
QUANTIZE PARAMETERS
REDUCE REFLECTION COEFFICIENTS TO PREDICTION COEFFICIENTS
MOVE DATA DOWNWARDS
RUN Q LOOP AND QUANTIZE
CPACK
PACK PARAMETERS
SIMULATE ERRORS ?
NO
YES
BURSTD
SIMULATE ERRORS
100 FRAMES SINCE PRINT
NO
YES
PRINT TOTAL NUMBER OF ERRORS
INVERT
INVERT BITS IN ERROR
UNPACK
UNPACK PARAMETERS
LOOK-UP QUANTIZER LEVEL
REDUCE REFLECTION COEFFICIENTS TO PREDICTION COEFFICIENTS
RECONSTRUCT SIGNAL
CLIP OUTPUT
OUTPUT DATA

FIGURE 2.4   INTEGER SIMULATION OF APC

2.3 EXPERIMENTAL RESULTS

Although various signal-to-noise ratio measurements were made on the chosen system (see Section 2.4), a much more meaningful measurement of performance was obtained through standardized intelligibility tests performed at Fort Huachuca. These tests were used to determine the intelligibility in the presence of channel errors as well as the intelligibility over an error free channel. The channel simulation used is discussed in Appendix A. Figure 2.5 shows the results obtained for the four channels. As can be seen, error rates as high as 10^{-3} have virtually no effect on intelligibility.

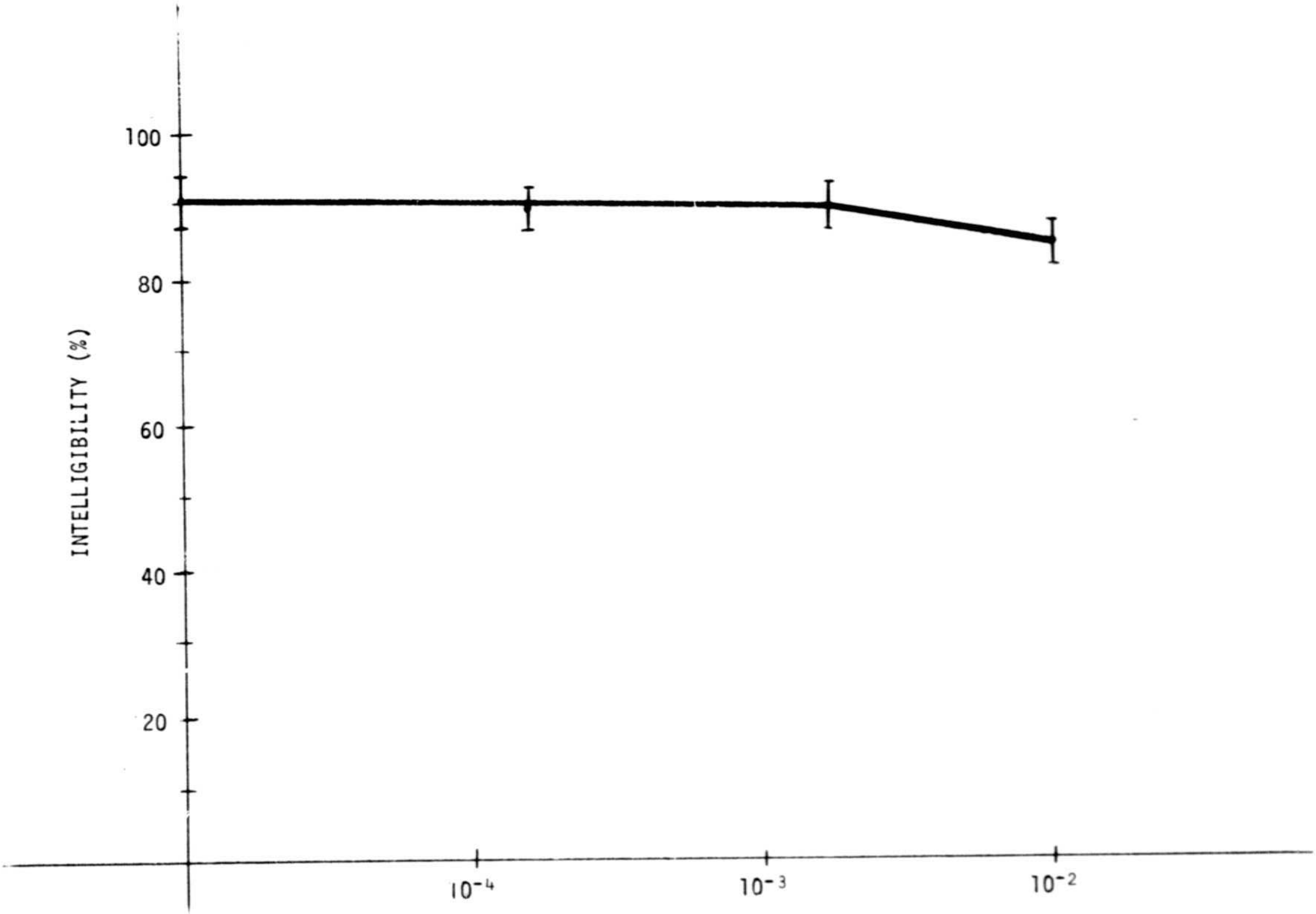
100
80
60
40
20
INTELLIGIBILITY (%)
10^{-4}
10^{-3}
10^{-2}
BIT ERROR RATE (BURST ERRORS)

Figure 2.5

2.4 TRADEOFFS

In order to optimize the chosen system, several parameters were varied and the signal-to-noise ratio in the transmitter loop was measured on a frame basis. The SNR measurement is defined as

$$\text{SNR} = 10 \log_{10} \left[\frac{\text{Input Signal Power}}{\text{Quantization Noise Power}} \right]$$

The quantization noise power and the input signal power were each averaged over a frame, and then the SNR for that frame was computed as above. The parameters examined were frame length, number of coefficients, bit allocation (quantization) of coefficients, and the quantizer level scale factor.

The performance was evaluted by examining the signal-to-quantizer noise ratios on 0.6 seconds of speech. The input SNR of the digitized speech was estimated at 30 to 40 dB.

Three frame sizes were examined and the results are shown in Figure 2.6 For the most part. the shortest frame (15 ms) gave slightly higher SNRs on most of the samples. In one region of very rapid transition, the 20 ms frame gave better results than either the 15 or 25 ms frames. This was probably due to the particular alignment of each of the frame boundaries at that instant. Listening indicated no perceptual difference between the three variants.

Figure 2.7 shows the results of comparing three systems: one with 3 coefficients, one with 4, and one with 6. Both the SNRs and listening indicated slight degradation with the use of 3 coefficients, and virtually no difference between the systems with 4 and 6 coefficients. In all three cases each coefficient was quantized to 5 bits.

Since little is gained by using more than 4 coefficients, we next examined bit allocations for 4 coefficients. A total of 16 bits were allocated in four different ways:

Allocation	K_1	K_2	K_3	K_4
1	4	4	4	4
2	5	4	4	3
3	5	5	4	2
4	5	5	3	3

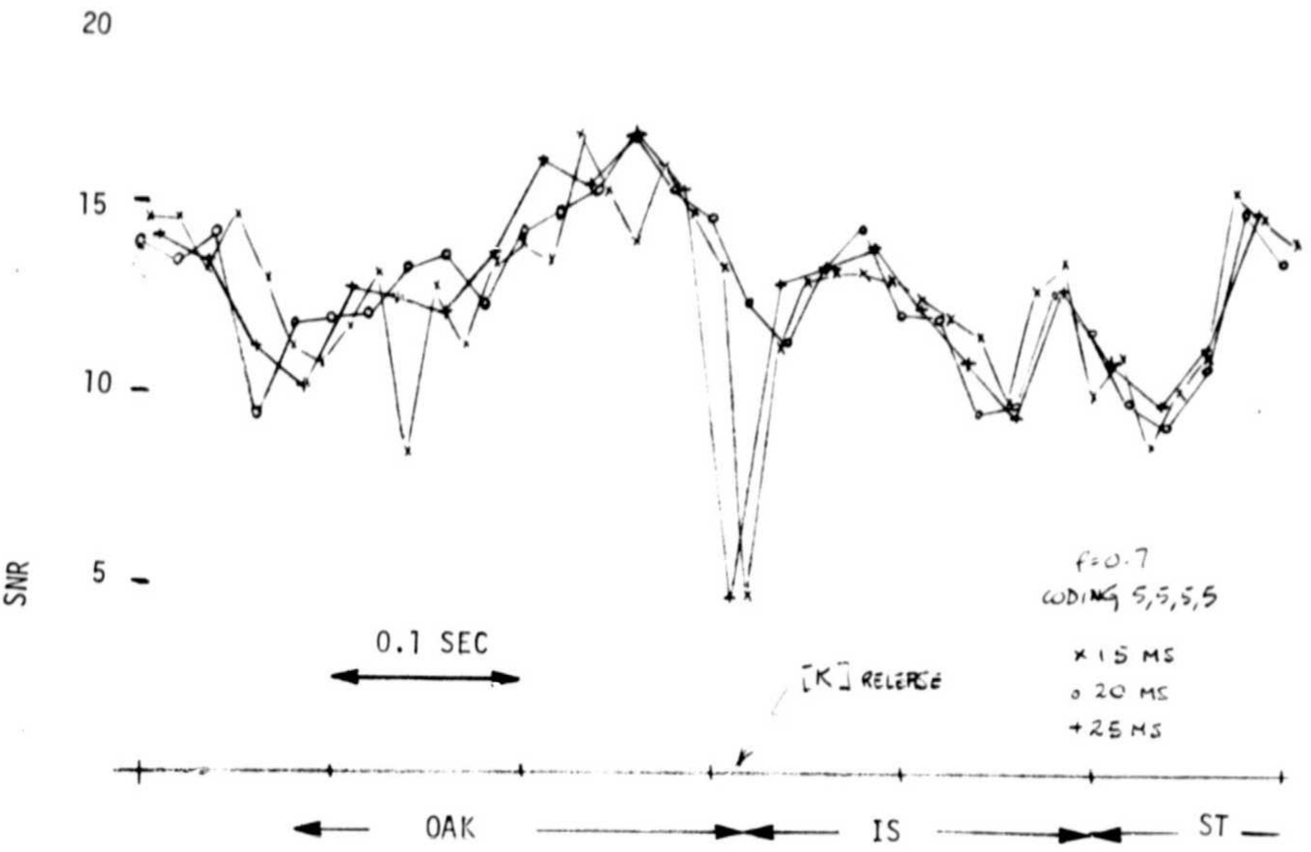

Figure 2.6

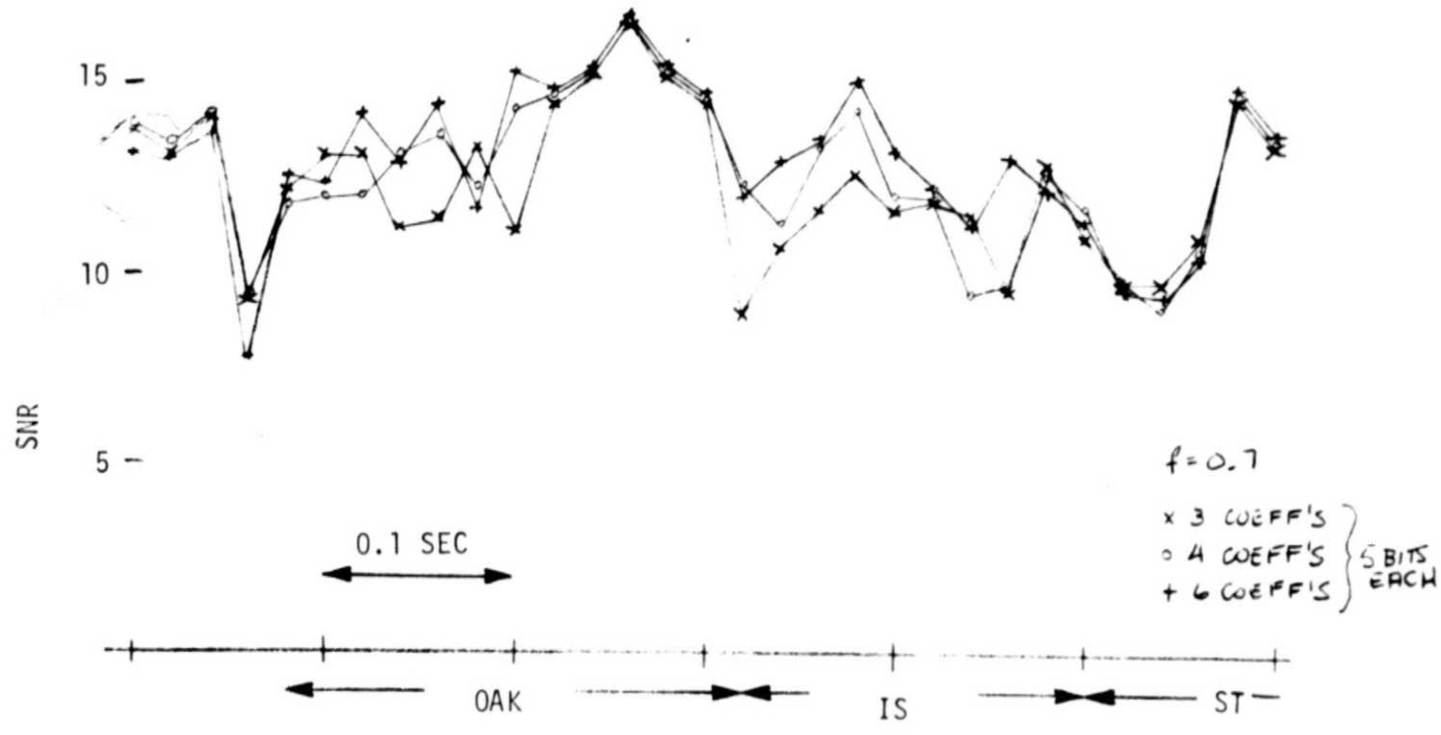

Figure 2.7

The first allocation was judged to be slightly worse than the remaining three which were all roughly equivalent. Figure 2.8 gives the results for the first and fourth allocations.

The final parameter examined was a scale factor used in setting the quantizer level Q, i.e., the value of Q before quantization was

$$Q = f \cdot F\ (\alpha/R_0)\ \sqrt{\alpha/N}$$

$F(\alpha/R)$ was taken as $(1 - .9 \frac{\alpha}{R_0})$. Thus, the theoretically optimum value of f is 0.73 for errors with Laplacian statistics and 0.82 for errors with Gaussian statistics. In Figure 2.9 the resulting SNRs are shown for four values of f ranging from 0.5 to 1.1. Although the SNRs indicate that the values 0.7 through 1.1 are roughly equivalent, and 0.5 is superior only on near silence intervals, in fact 0.5 was judged perceptually the most pleasing on all speech. This is probably due to a phenomenon which has been noted elsewhere[3], i.e., that the human ear tolerates the distortion due to slope overload more readily than quantizing noise. In further perceptual evaluation, values of f in the range 0.4 to 0.5 were found to be the most generally satisfactory.

As a result of the above evaluations, the following alternate baseline was established:

Sampling Rate:		7.0 KSPS
Frame Length:		25 ms
Parameter Encoding:		
	K_1	5 bits
	K_2	5 bits
	K_3	3 bits
	K_4	3 bits
	Q	4 bits
Total		20 bits
Synch:		5 bits

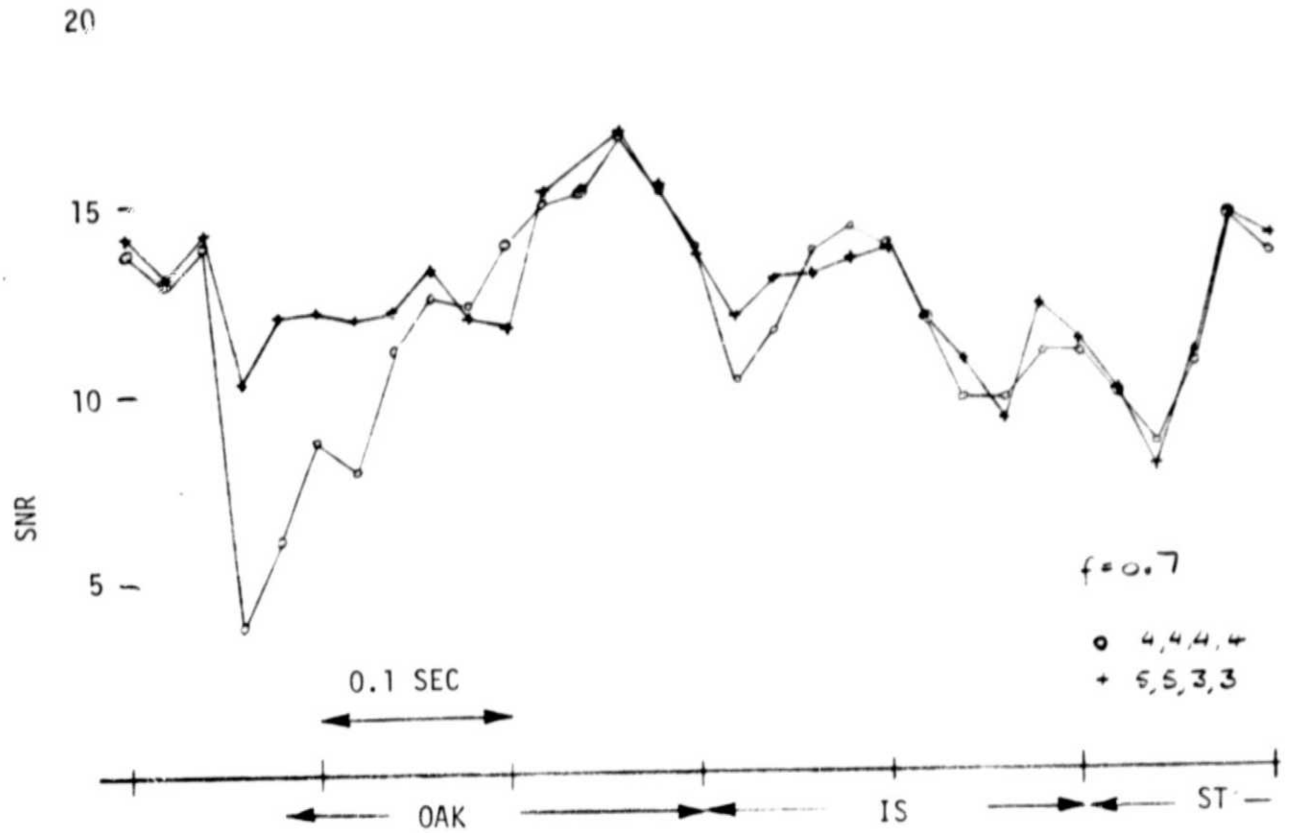

Figure 2.8

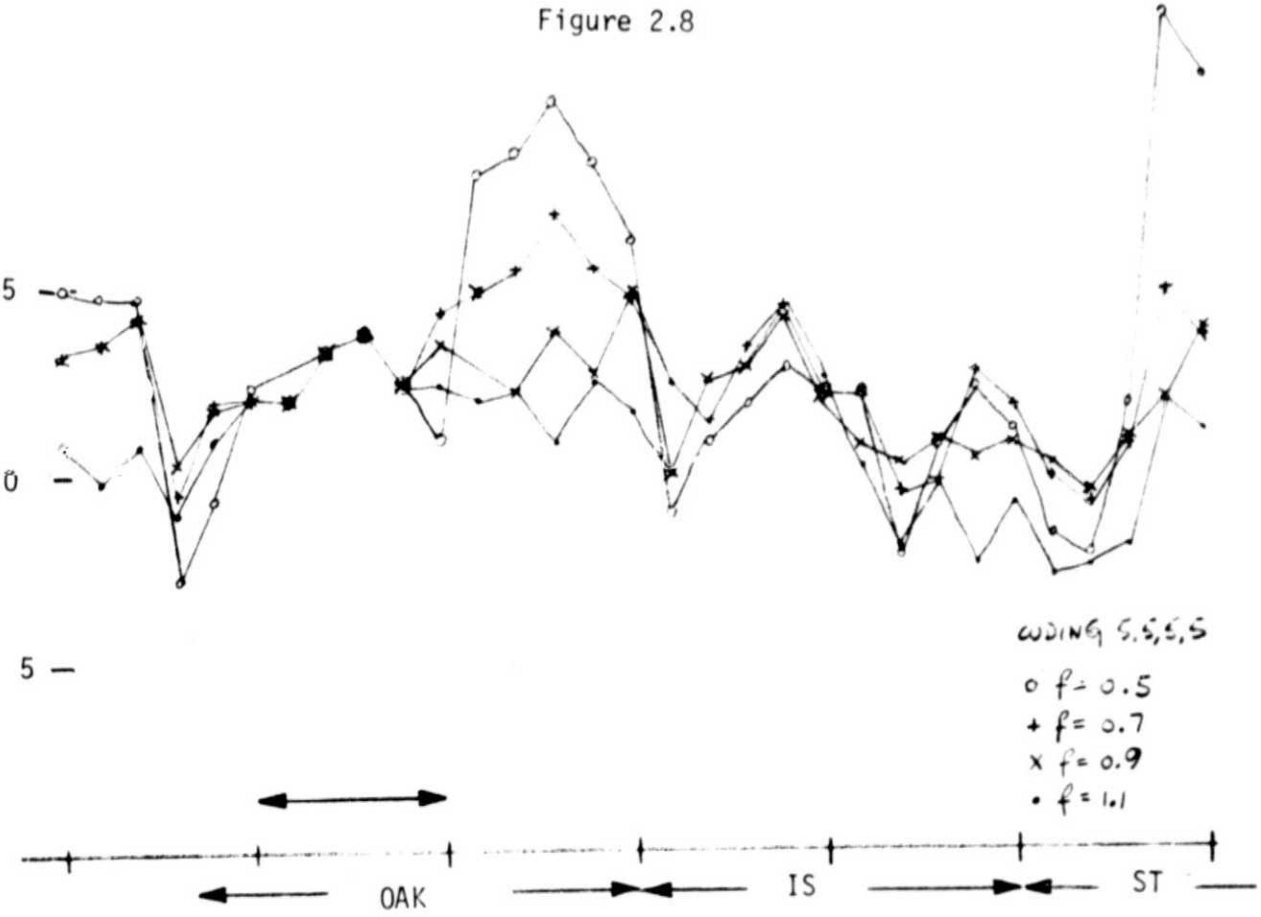

Figure 2.9

Parameter Rate:	800 bps
Synch Rate:	200 bps
Residue Rate:	7000 bps
Total Data Rate:	8000 bps

One tape, an FM news broadcast, was processed using both this system and the baseline used for the other tests. No significant perceptual differences were noticed between the two systems.

The conclusion we reach is that there is substantial flexibility in the allocation of bandwidth in the neighborhood of the original baseline allocation. Minor changes resulting, for example, from the inclusion of either synch or error protection a few parameters would not significantly influence received voice quality.

3. IMPLEMENTATION

3.1 NETWORK SIGNAL PROCESSOR INTERFACE

The two voice compression modems utilized in the orbiter NSP are shown in Figure 3.1. The analysis and synthesis functions of each modem are included in a single package, called a 1-AU, to allow sharing of common processing elements. Within the NSP, the voice compression units interface directly with the TDM MUX/DEMUX equipment where the compressed voice is exchanged. Timing information for external framing and clocking the digitized voice into and out of the voice units is provided by the MUX/DEMUX units. Within the units frame synch for the voice algorithm can be provided, or the NSP may provide frame synch signals. The choice between these two options should be based on a study of the overall impact on the NSP hardware (see also Section 3.3). Each unit interfaces with the audio center via two analog voice signals: one from ground to orbiter, and one from orbiter to ground. The data shown are for 8 kbps APC, or the 32/24 kbps VSD currently baselined, which is shown parenthetically.

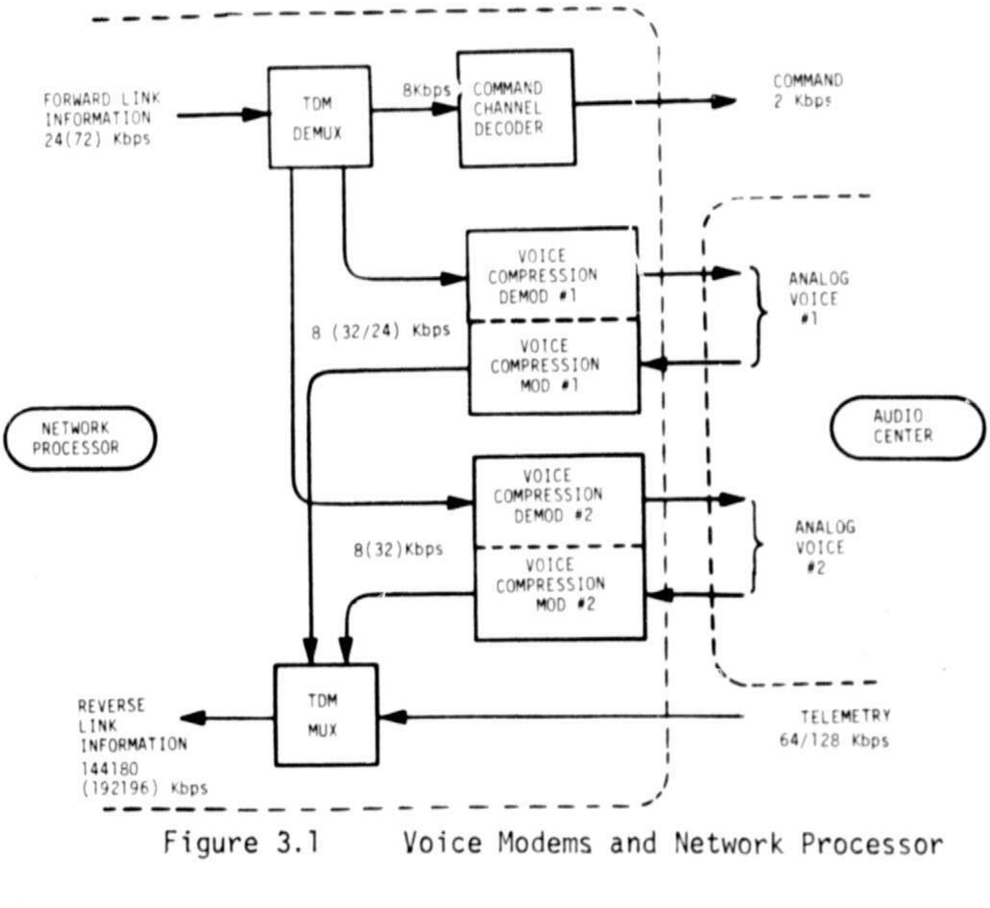

Figure 3.1 Voice Modems and Network Processor

3.2 NASA-APC IMPLEMENTATION ON THE 1-AU PROCESSOR

A full-duplex NASA-APC system has been designed for real-time implementation utilizing the 1-AU processor shown in Figure 3.2. A simplified block diagram of a hook-up between two voice processing systems is illustrated in Figure 3.3. The system is designed to start randomly when powered up and run continuously until powered down.

A preliminary coding exercise was performed to provide detailed timing estimates and hardware requirements. The timing estimates were developed on a 300 ns instruction execution time base, and are summarized in Table 3.1.

The 1-AU processor does not have normalize or divide instructions. The APC transmitter algorithms require 1 normalization and 5 divisions per frame. It was determined that these operations could be implemented in the existing software capability without additional hardware.

The APC algorithms used in the integer simulation on the Interdata 85 were converted to the 1-AU processor in a real-time frame-work augmented by the input/output operations necessary to the system. The sampling rate was modified from 7000 samples per second to 6750 samples per second to accommodate both APC data and synch in the 8000 bits per second communication channel, since this is a worst case configuration. The flowchart presented in Figure 3.4 provides a functional summary of the program.

The input/output of the NASA-APC transmitter is asynchronous to that of the receiver; however, the program will function synchronously. Current transmitter output and receiver input are buffered in the MUX buffers, while the next frames are being processed. The D/A outputs are double-buffered in the RAMs as are the A/D inputs. The latter are double-buffered in both RAMs to provide maximum efficiency during the construction of the autocorrelation matrix (i.e., both the multiplier and the multiplicand can be loaded with one instruction).

Notification of input/output status is via discretes. The execution of the APC algorithms is time-sliced accordingly to ensure proper servicing of the A/D and D/A sampling rate. At a sampling rate of 6750 samples per second, the program is structured to check the I/O status of the A/D and D/A within 148 micro time increments. In the event the A/D status indicates no input available, zeros will be processed.

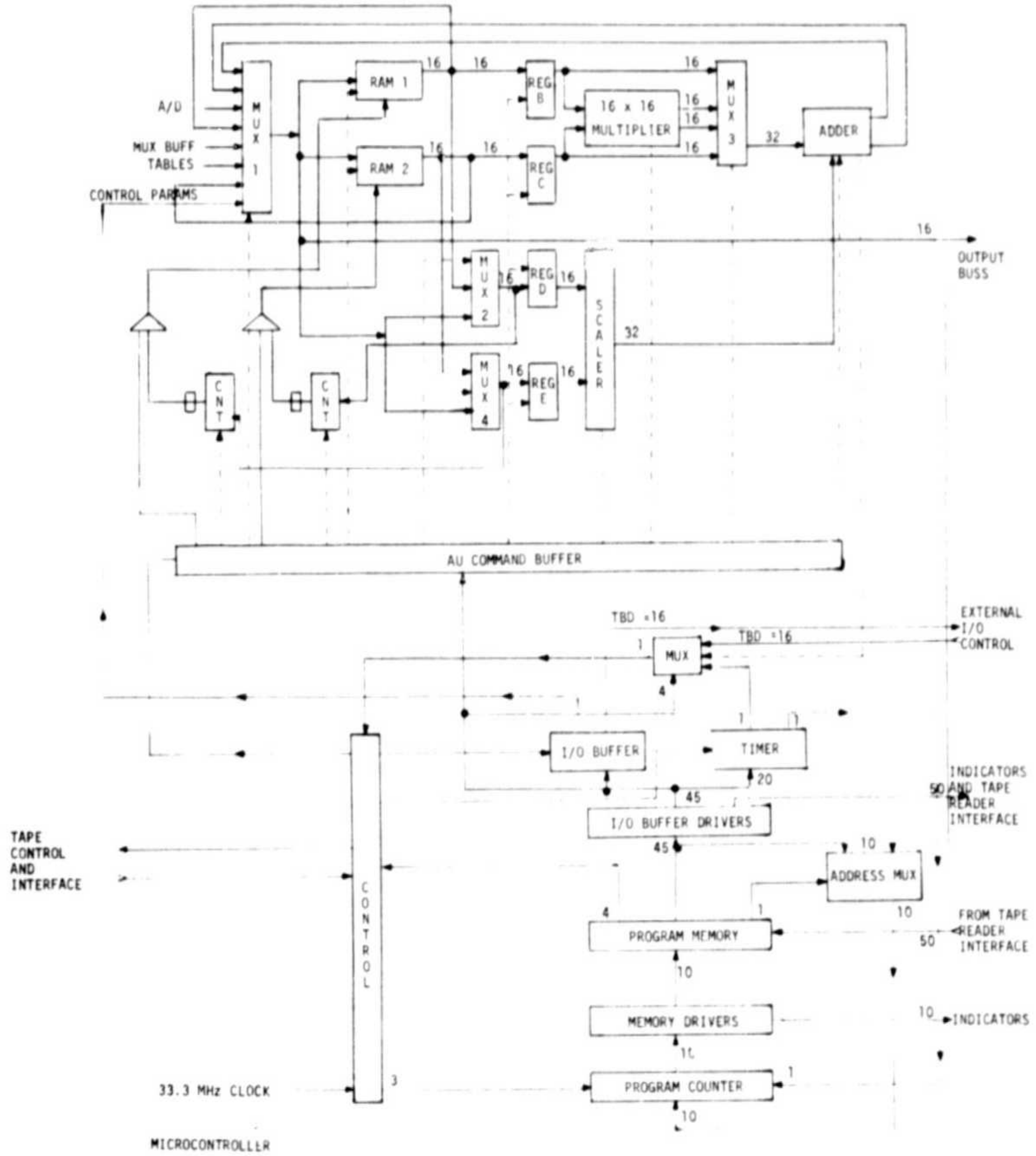

FIGURE 3.2 FUNCTIONAL BLOCK DIAGRAM OF 1-AU PROCESSOR

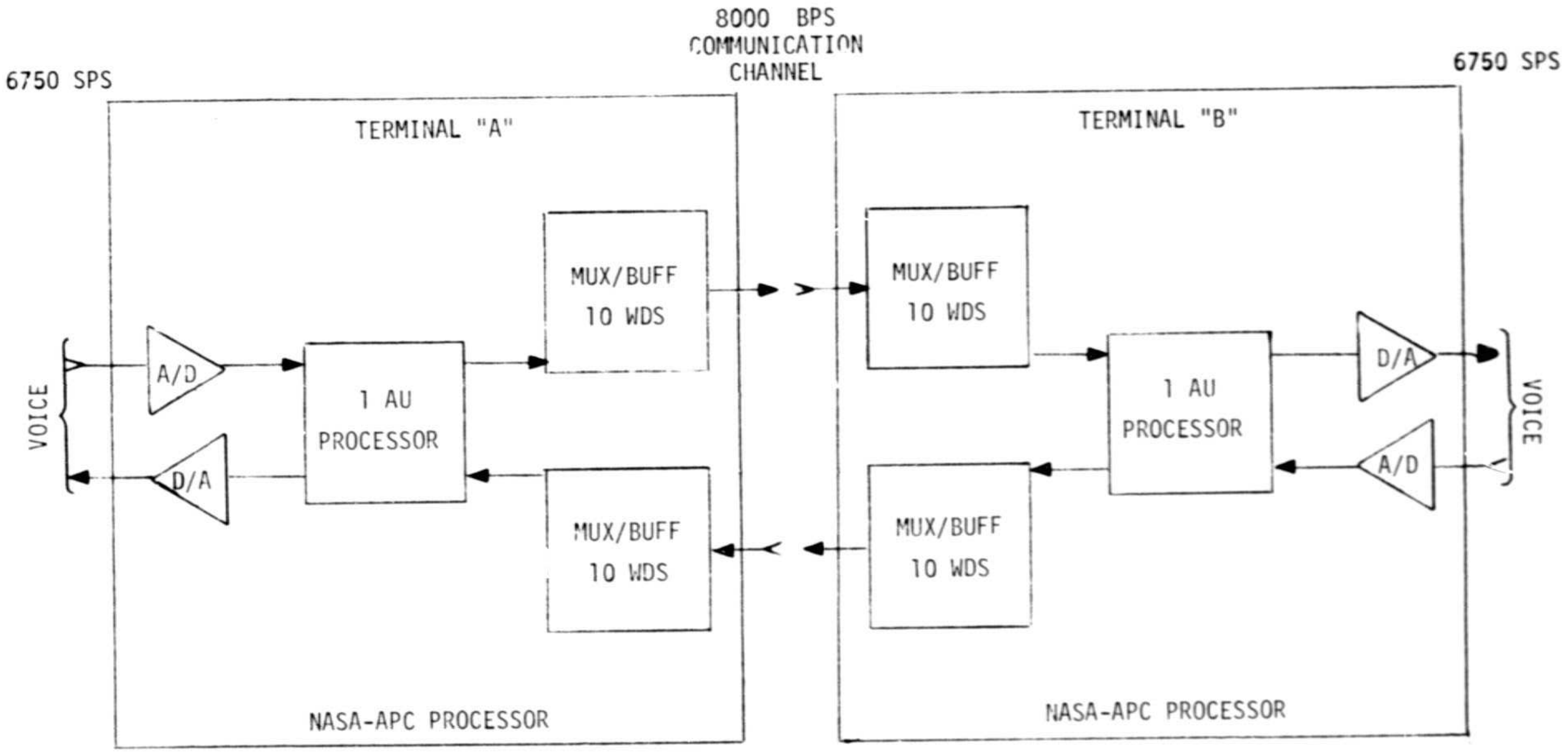

Figure 3.3 Simplified Block Diagram of a Two Terminal NASA-APC Full Duplex Hookup

Table 3.1

NASA-APC Implementation on 1-AU Processor
Timing and Sizing Estimates

Function	Op.Cycles	RAM	ROM
Construct Autocorrelation Matrix and Apply Stability Weighting.	3553	146	123
Invert Autocorrelation Matrix	372	7	132
Calculate Quantizer Level	136	22	66
Quantize Reflection Coefficients	16	4	[illegible]
Convert Reflection Coefficients to Prediction.	26	4	26
Execute Predictor Filter	3809	145	229
Pack Transmitter Output Buffer	945	10	34
Unpack Receiver Input Buffer	945	10	34
Determine Quantizer & Predictor Coefficients.	30	-	30
Reconstruct Signals	2185	145	85
Clip D/A Output	407	-	6
Refresh RAM Constants	360	-	6
A/D, D/A Input/Output	540	540	540
Service MUX Buffers	40	-	10
Housekeep RAM Buffers	280	-	10
TOTAL	13644	1033	1347

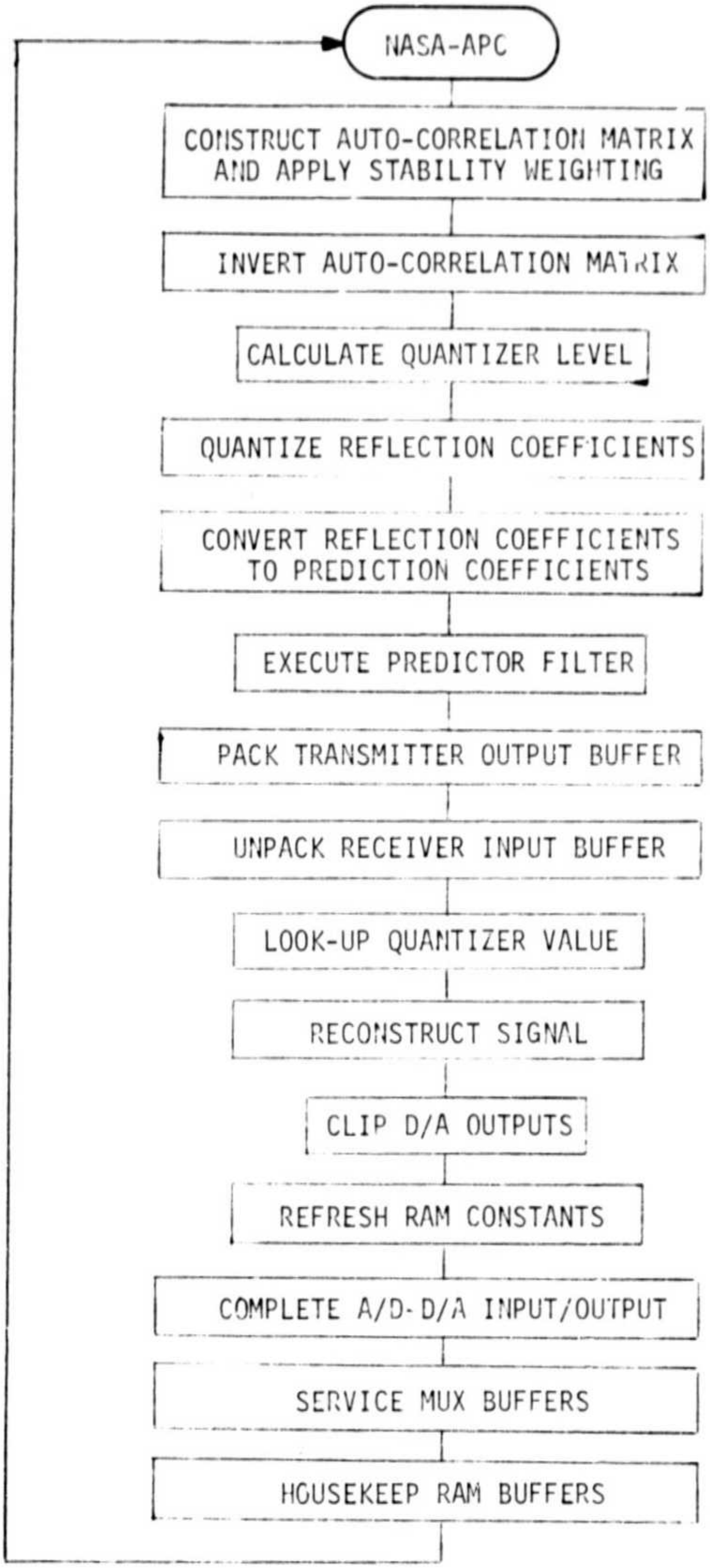
NASA-APC
CONSTRUCT AUTO-CORRELATION MATRIX AND APPLY STABILITY WEIGHTING
INVERT AUTO-CORRELATION MATRIX
CALCULATE QUANTIZER LEVEL
QUANTIZE REFLECTION COEFFICIENTS
CONVERT REFLECTION COEFFICIENTS TO PREDICTION COEFFICIENTS
EXECUTE PREDICTOR FILTER
PACK TRANSMITTER OUTPUT BUFFER
UNPACK RECEIVER INPUT BUFFER
LOOK-UP QUANTIZER VALUE
RECONSTRUCT SIGNAL
CLIP D/A OUTPUTS
REFRESH RAM CONSTANTS
COMPLETE A/D-D/A INPUT/OUTPUT
SERVICE MUX BUFFERS
HOUSEKEEP RAM BUFFERS

Figure 3.4 Functional Flow Chart of Frame Processing of NASA-APC Implemented on 1 AU Processor

When the APC algorithms have been executed, the remaining frame time is spent completing the A/D and D/A input/output, switching buffer references, and housekeeping the MUX buffers. If the status of the receiver MUX buffer indicates no data available, zeros will be processed.

3.3 FLIGHT HARDWARE CONFIGURATION

The functional configuration assumed for flight operation is shown in Figure 3.5. The framing shown corresponds to the baseline described in Sections 2.1.4 and 3.2. The APC analysis and synthesis would be accomplished on a 1-AU unit as described in Section 3.2.

A preliminary IC count including MSI and LSI is:

Frame Synch & NSP Interface	20 ICs
Serial/Parallel & Parallel/Serial	6 ICs
1-AU	158 ICs
A/D, D/A & Buffers	10 ICs
Total	194 ICs

Since the sizing in Section 3.2 indicates 20% busy and 80% dead time at a 300 ns cycle time, a cycle time of 1 μsec would yield a 68% busy status. This would permit implementation in C-MOS. Figuring from a 100 mw per chip average power dissipation for C-MOS, this gives a total power consumption of roughly 20 watts per full duplex voice unit. At standard flight hardware packing densities of 55 IC per 20.32 cm x 15.24 cm (8"x6") board, four boards would be required. These four boards may be mounted in a module, roughly 5.08 cm x 15.24 cm x 20.32 cm (2"x6"x8") weighing 1.1 kg (2-1/2 lbs.).

The only effect of removing frame synch, i.e., assuming it is provided by the NSP, is to decrease the chip count to 175 ICs.

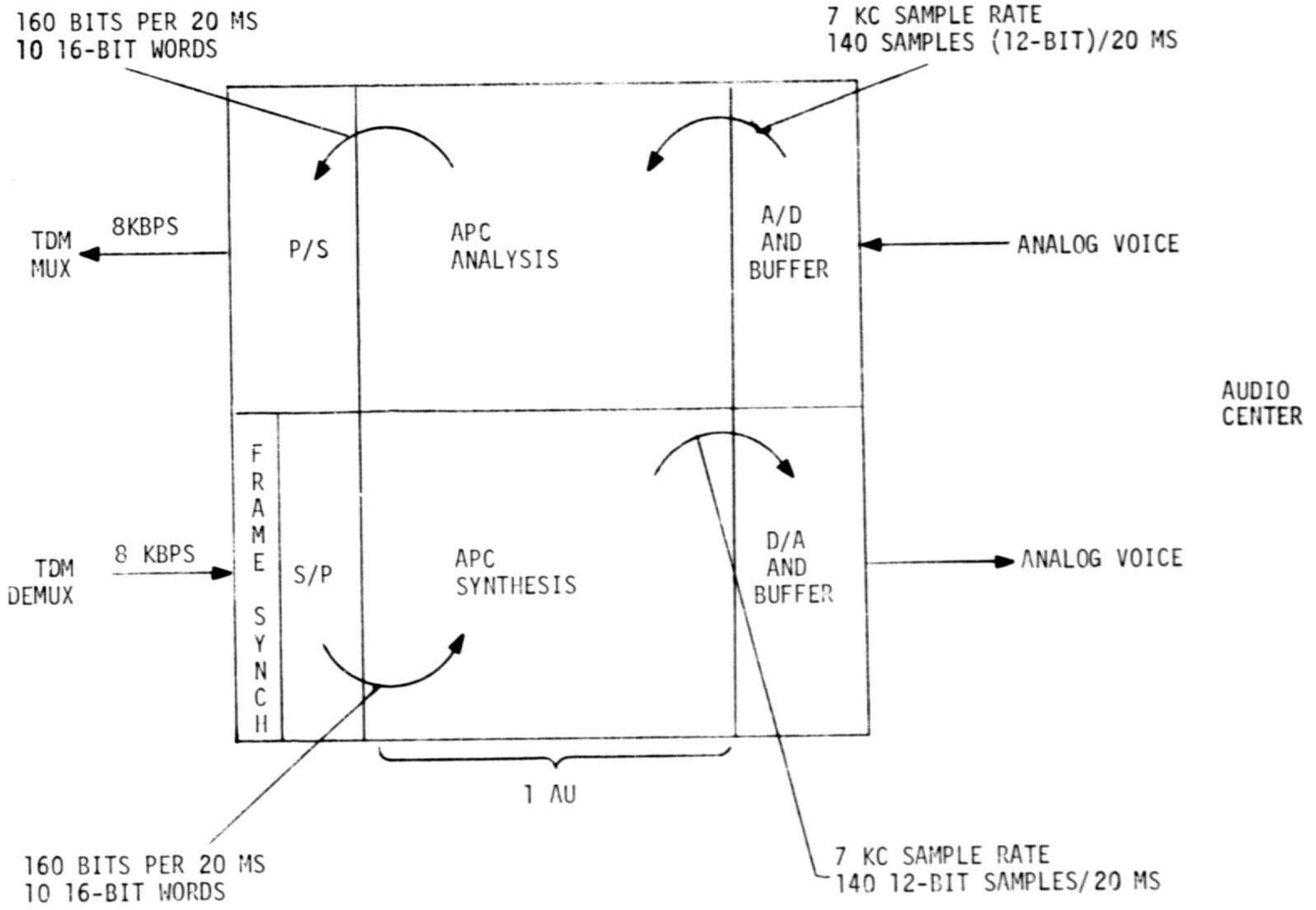
160 BITS PER 20 MS
10 16-BIT WORDS
7 KC SAMPLE RATE
140 SAMPLES (12-BIT)/20 MS
TDM
MUX
8KBPS
P/S
APC
ANALYSIS
A/D
AND
BUFFER
ANALOG VOICE
AUDIO
CENTER
FRAME SYNCH
TDM
DEMUX
8 KBPS
S/P
APC
SYNTHESIS
D/A
AND
BUFFER
ANALOG VOICE
1 AU
160 BITS PER 20 MS
10 16-BIT WORDS
7 KC SAMPLE RATE
140 12-BIT SAMPLES/20 MS

Figure 3.5

4. RECOMMENDATIONS

4.1 BASELINE

The following allocation represents a base point about which variations are possible to accommodate synch, error protection, or minor changes in the NSP baseline:

Sampling rate	7 kbps
Frame rate	50 Hz
Parameters:	
K_1:	5 bits
K_2:	5 bits
K_3:	3 bits
K_4:	3 bits
Q :	4 bits
RESIDUE:	7 kbps
PARAMETERS:	1 kbps
TOTAL DATA RATE:	8 kbps

4.2 FUTURE WORK

Tests conducted on similar systems have indicated that the received voice quality is particularly sensitive to errors in certain of the parameter bits. Thus, an obvious possibility which should be examined is the allocation of a portion of the allowed data rate to error correction. This could best be accomplished by first determining the relative sensitivity of the received voice to errors in each of the parameter bits and then applying coding beginning with the most sensitive bits and proceeding toward the least sensitive until an optimum is found for a typically noisy channel. Since the optimum split between information and redundancy rates is probably dependent on channel characteristics, different techniques might be used on the uplink and downlink.

Another area worthy of investigation is the portion of the algorithm which sets the quantizer level. Results obtained so far indicate that this is the most sensitive part of the algorithm, and that an optimum technique has yet to be developed. A systematic examination of the effects of the quantizer level on various signal types is needed to derive a truly optimum setting. Once this optimum setting is derived the technique would be refined to provide noise squelch specific to the orbiter environment.

REFERENCES

[1] B. S. Atal & S. L. Hanauer, "Speech Analysis & Synthesis by Linear Prediction of the Speech Wave", JASA, Vol. 50 No. 2, p.2, 1971 .

[2] J. D. Markel & A. H. Gray, "On Autocorrelation Equations as Applied to Speech Analysis", IEEE Trans. on Audio & Electro-Acoustics, Vol. AV-21, pp 69-79, April 1973.

[3] N. S. Jayant & A. E. Rosenberg, "The Preference of Slope Overload to Granularity in the Delta Modulation of Speech", BSTJ, Vol. 50, #10, pp 3117-3125, December 1971.

APPENDIX A. SIMULATION OF CHANNEL ERRCR STATISTICS

It was assumed that the APC bit stream would be transmitted using an optimum rate 1/3, constraint length 7 convolutional code. Dr. Gaylord Huth of Axiomatix provided an algorithm for generation of upper bounds on the probability of burst of errors as a function of E_s/N_o and burst length. The probability of a burst of length b can be upper bounded by

$$Q_b = \sum_{k=k_{b,\ min}}^{K_{b,\ max}} n_{bk}\ P_k$$

where P_k is the probability of incorrectly choosing a weight k code word over the all zeroes code word, and $n_{b,\ k}$ is the number of weight k code words caused by bursts of length b. P_k is given by

$$P_k = \int_{\sqrt{2kE_s/N_o}}^{\infty} \frac{e^{-t^2/2}}{\sqrt{2\pi}}\ dt = Q\left(\sqrt{\frac{2kE_s}{N_o}}\right)$$

n_{bk} depends on the code used and there is no general formula for computing it. For the code assumed here

$$n_{1,\ j} = \begin{cases} 1 & j = 14 \\ 0 & \text{else} \end{cases}$$

$$n_{2,\ j} = \begin{cases} 1 & j = 16 \\ 0 & \text{else} \end{cases}$$

The Fortran version of an algorithm supplied by Axiomatix is shown in Figure A.1. This program was used to generate values of $n_{b,k}$ for $b > 2$, and Q_b for all values of b.

The total probability of error at the output of the Viterbi decoder can

```
        C  SOURCF IS IN CODFIN SU 1.0
0004R          INTEGER*2 IS(200)
0004R          DIMENSION IW(200)
0004R          DATA IS/200*0/
019CR          ISUM=0
01A0R          IADC=0
01A6R          IAN=1
01F0R          ISTART=1
01E0R          WRITE(5,100)
01D0R   100    FORMAT('  ENTER ES/NO   ,F7.5')
01FFR          READ(5,101)XX
020ER   101    FORMAT(F7.5)
0216R          CALL REJECT
021FR          WRITE(3,102)XX
023ER   102    FORMAT(1X,'*********** ES/NO= ',F7.5,' **********')
026ER          XX=2.*XX
027AR          P111=DF(SQRT(14.*XX))
029ER          IB=1
02A6R          P222=DF(SQRT(16.*XX))
02CAR          WRITE(3,7)IB,P111
02F2R          IB=2
02FAR          WRITE(3,7)IB,P222
0322R          P2=0
032AR          P2=1.5*P111
0336R          P2=P2+2.*P222
0346R          IS(7)=1
0352R          CALL RSHFTD(ISTART,16)
035CR          IZERO=0
0364R          CALL RSHFTD(IZERO,16)
036ER          ISIX=6
0376R          CALL RSHFTD(ISIX,16)
0380R          DO 1 IB=3,20
038ER          WRITE(3,6)
03A0R   6      FORMAT('0','    B L       WEIGHT     NO OF CODWRDS')
03D2R          P=0
03DAR          DO 10 I=1,200
03E2R          CALL EQU4(IZERO,IW(I))
0400R   10     CONTINUE
0412R          IB2=IB-2
041ER          IS(IB+6)=1
        C
        C  NN IS TH N-1 COUNT
0436R          CALL EQU4(IZERO,NN)
        C  NB=2**(IB2)
0440R          CALL EQU4(ISTART,NB)
044AR          CALL LSHFTD(NB,IB2)
        C
        C  J1 IS THE COUNT DOWN INDEX
0454R          CALL EQU4(NB,J1)
        C  KK=N-1
045ER          CALL EQU4(IZERO,KK)
0468R   2      CALL ISUBDP(J1,J1,ISTART)
        C  IF NEGATIVE  LOOP DONE
0474R          IF(J1)30,31,31
0484R   31     DO 20 I=1,IB2
048CR          IS(I+7)=0
        C  J=2**(IB2-I)
04A4R          CALL EQU4(ISTART,J)
04AER          J3=IB2-I
04BAR          CALL LSHFTD(J,J3)
        C  IF(KK.LT.J)GO TO 20
04C4R          CALL EQU4(KK,KK1)
04CER          CALL ISUBDP(KK1,KK1,J)
04DAR          IF(KK1)20,21,21
04EAR   21     IS(I+7)=1
        C  KK=KK-J
0502R          CALL EQU4(KK1,KK)
        C
050CR   20     CONTINUE
051ER          IB6=IB+6
052AR          ISUM=0
0532R          DO 3 I=1,IB6
053AR          III=I+6
0546R          ISUMS=IS(III)+IS(I+5)+IS(I+3)+IS(I+2)+IS(I)
05A2R          ISUM1=0
05AAR          IF(IAND(ISUMS,IAN))111,111,112
05C2R   112    ISUM1=1
05CAR   111    ISUMS=IS(III)+IS(I+3)+IS(I+2)+IS(I+1)+IS(I)
0626R          ISUM2=0
062ER          IF(IAND(ISUMS,IAN))222,222,223
0646R   223    ISUM2=1
064ER   222    ISUMS=IS(III)+IS(I+4)+IS(I+1)+IS(I)
0696R          ISUM3=0
069ER          IF(IAND(ISUMS,IAN))3,3,334
06D6R   334    ISUM3=1
06DER   3      ISUM=ISUM+ISUM1+ISUM2+ISUM3
06E4R          J=ISUM/2-6
06FAR          CALL IADDDP(NN,NN,ISTART)
0706R          CALL EQU4(NN,KK)
0710R          CALL IADDDP(IW(J),IW(J),ISTART)
0744R          GO TO 2
074CR   30     MM=0
0750R          CALL EQU4(IZERO,JSUM)
075AR   4      MM=MM+1
0766R          M=(MM+6)*2
0776R          C=FLOAT4(IW(MM))
0796R          WRITE(3,5)IB,M,C
07CCR   5      FORMAT(I7,I10,E20.12)
07D4R          P=P+C*DF(SQRT(M*XX))
0810R          CALL IADDDP(JSUM,JSUM,IW(MM))
0820R          CALL EQU4(JSUM,IADC)
083AR          CALL ISUBDP(IADC,IADC,NB)
0846R          IF(IADC)4,40,40
[illegible]   40     WRITE(3,7)IB,P
[illegible]   7      FORMAT(10X,'P(',I2,')= ',E15.5)
[illegible]          PB=((FLOAT(IB)/2.)+1.)*P   → PB
[illegible]          P2=P2+PB
[illegible]   1      CONTINUE
[illegible]          WRITE(3,8)P2
[illegible]   8      FORMAT(10X,' P2= ',E15.5)
[illegible]          END
```

FIGURE A.1

be upper bounded by

$$P_e < Q_1 + \sum_{b=2}^{\infty} Q_b(\frac{b-2}{2} + 2)$$

This upper bound assumes that a burst begins and ends with a "1" and contains a random binary sequence between the two end points. In fact, a burst as defined for this code contains no more than six consecutive zeroes at any point internal to the burst. Thus, P_e as given above is upper bounded in both Q_b and its weighting factor. An approximation to P_e is also computed by the program in Figure A.1 as

$$\hat{P}_e = Q_1 + \sum_{b=2}^{N} Q_b(\frac{b-2}{2} + 2)$$

where N is the maximum value of b used.

Due to the excessive execution time of the program for burst lengths longer than 20, it was not practical to compute values of Q_b beyond that point. However, for $E_s/N_o \leq 0.6$ [i.e., $P_e \geq 10^{-3}$], bursts of these longer lengths are nearly as probable as shorter bursts. Efforts to approximate the probabilities of longer bursts were not successful, hence the curves for Q_b versus b were extrapolated to larger values of b until the value of $\hat{P}_e$ was approximately the predicted error rate. Table A.1 gives the values of Q_b, $\hat{P}_e$, and the probability of error observed in the simulation for $E_s/N_o = 0.52$, 0.6, and 0.7. Figure A.2 shows calculated and extrapolated values of Q_b.

The routine for simulating the burst errors acts as a two state device where the first state represents the generation of a string of zeroes and the second state represents the generation of a burst. The burst generation state is entered with probability $P = \sum_{b=1}^{N} Q_b$ where N is the maximum burst length being used. The distance between bursts is a random variable uniformly distributed on (0, 2/P). Thus, the average distance between bursts is 1/P. When the burst generation state is entered, a burst length, J, is randomly chosen according to the conditional distribution $\{Q_b/P\}$. Then a "1" is generated followed by J-2 "0"s and "1"s which are equally likely unless six "0"s have been generated in succession, at which time a "1" is forced.

Table A.1 Calculated and Observed Probabilities

	E_s/N_o = .52	E_s/N_o = .6	E_s/N_o = .7
b	Q_b	Q_b	Q_b
1	$.68 \times 10^{-4}$	$.25 \times 10^{-4}$	$.59 \times 10^{-5}$
2	$.23 \times 10^{-4}$	$.60 \times 10^{-5}$	$.11 \times 10^{-5}$
3	$.30 \times 10^{-4}$	$.76 \times 10^{-5}$	$.14 \times 10^{-5}$
4	$.71 \times 10^{-4}$	$.18 \times 10^{-4}$	$.35 \times 10^{-5}$
5	$.65 \times 10^{-4}$	$.16 \times 10^{-4}$	$.29 \times 10^{-5}$
6	$.40 \times 10^{-4}$	$.82 \times 10^{-5}$	$.12 \times 10^{-5}$
7	$.44 \times 10^{-4}$	$.83 \times 10^{-5}$	$.11 \times 10^{-5}$
8	$.56 \times 10^{-4}$	$.10 \times 10^{-4}$	$.14 \times 10^{-5}$
9	$.47 \times 10^{-4}$	$.78 \times 10^{-5}$	$.96 \times 10^{-6}$
10	$.51 \times 10^{-4}$	$.80 \times 10^{-5}$	$.95 \times 10^{-6}$
11	$.47 \times 10^{-4}$	$.65 \times 10^{-5}$	$.67 \times 10^{-6}$
12	$.48 \times 10^{-4}$	$.61 \times 10^{-5}$	$.59 \times 10^{-6}$
13	$.47 \times 10^{-4}$	$.54 \times 10^{-5}$	$.50 \times 10^{-6}$
14	$.46 \times 10^{-4}$	$.48 \times 10^{-5}$	$.41 \times 10^{-6}$
15	$.46 \times 10^{-4}$	$.44 \times 10^{-5}$	$.35 \times 10^{-6}$
16	$.45 \times 10^{-4}$	$.39 \times 10^{-5}$	$.30 \times 10^{-6}$
17	$.46 \times 10^{-4}$	$.35 \times 10^{-5}$	$.18 \times 10^{-6}$
18	$.45 \times 10^{-4}$	$.33 \times 10^{-5}$	
19	$.44 \times 10^{-4}$	$.30 \times 10^{-5}$	
20	$.44 \times 10^{-4}$	$.27 \times 10^{-5}$	
21	$.43 \times 10^{-4}$	$.25 \times 10^{-5}$	
22	$.43 \times 10^{-4}$	$.22 \times 10^{-5}$	
23	$.42 \times 10^{-4}$	$.20 \times 10^{-5}$	
24	$.42 \times 10^{-4}$	$.17 \times 10^{-5}$	
25	$.42 \times 10^{-4}$	$.15 \times 10^{-5}$	
26	$.41 \times 10^{-4}$	$.13 \times 10^{-5}$	
27	$.41 \times 10^{-4}$	$.12 \times 10^{-5}$	
28	$.40 \times 10^{-4}$	$.11 \times 10^{-5}$	
PREDICTED ERROR RATE	1.0×10^{-2}	$.92 \times 10^{-3}$	$.82 \times 10^{-4}$
ACTUAL ERROR RATE	1.0×10^{-2}	1.7×10^{-3}	1.62×10^{-4}

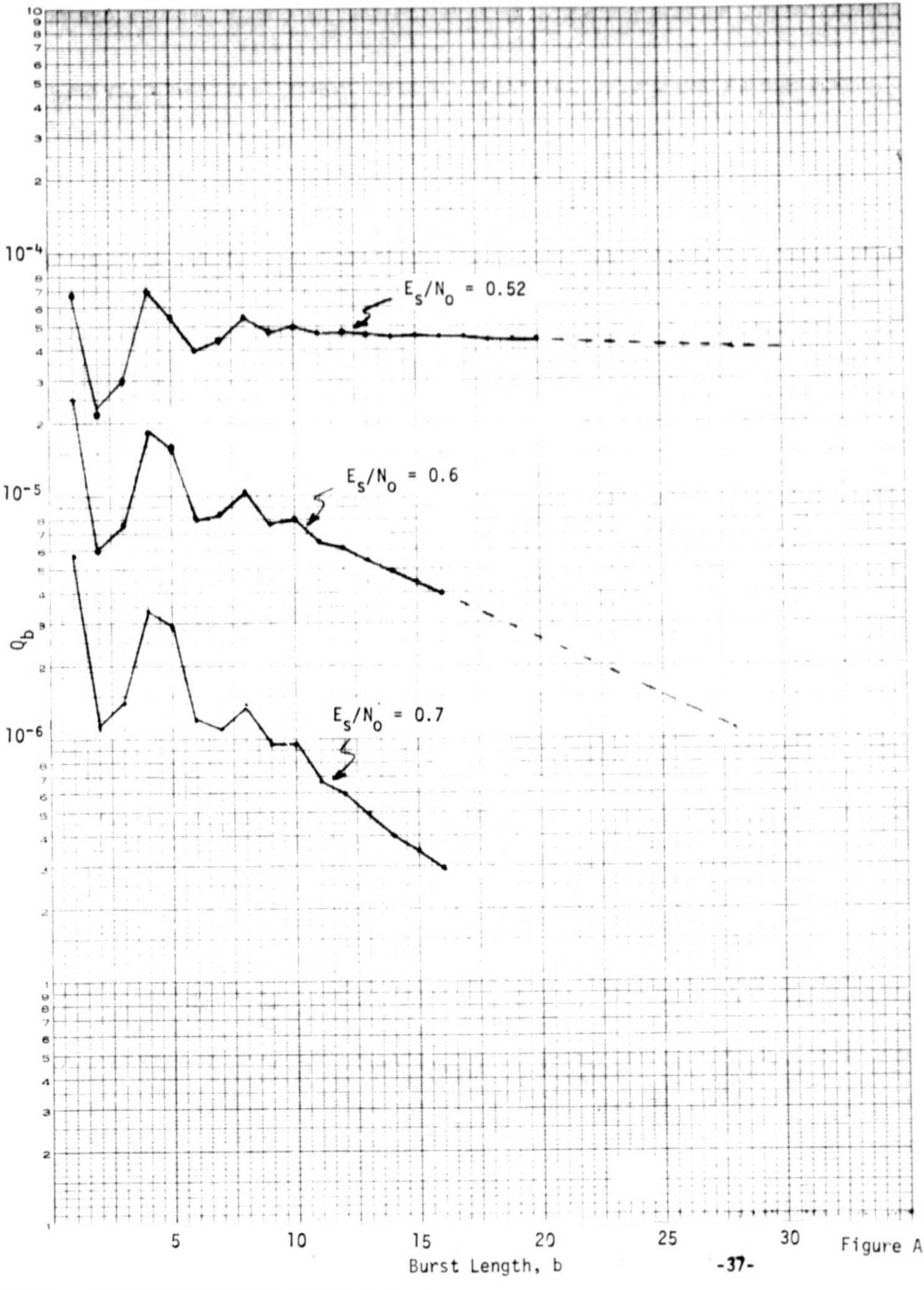
E_s/N_o = 0.52
E_s/N_o = 0.6
E_s/N_o = 0.7
Q_b
10^{-4}
10^{-5}
10^{-6}
Burst Length, b
5
10
15
20
25
30

Figure A.2

The Jth bit in the burst is forced to be a "1". After the Jth bit has been generated the routine returns to the zero generation state